Package 'notifyme'

October 13, 2022

Type Package

Title Send Alerts to your Cellphone and Phillips Hue Lights

Version 0.3.0

Maintainer James Black <james@epijim.uk>

Description Functions to flash your hue lights, or text yourself, from R. Designed to be used with long running scripts.

License GPL (>= 2)

URL https://github.com/epijim/notifyme

BugReports https://github.com/epijim/notifyme/issues

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

Imports magrittr, dplyr, httr

NeedsCompilation no

Author James Black [aut, cre, cph]

Repository CRAN

Date/Publication 2016-11-27 00:11:31

R topics documented:

create_hue_username *Create a username to enable HTTP requests on your hue hub*

Description

You need a username to access the hub. This function will create a username. Immediately before running this function you need to press the link button on the bridge, the big button on the Generation 2 hub, to prove you have access to this hub.

Usage

```
create_hue_username(bridge_ip = NULL)
```

Arguments

bridge_ip	The internal ip address of your hue bridge

Bugs

Code repo: https://github.com/epijim/notifyme

Examples

```
## Not run: (create_hue_username(bridge_ip)
```

get_hue_ip *Get the internal IP address of your Hue bridge*

Description

This function uses the bridge UPNP service to get the internal IP address of your hue hub. You need to be on the same network, else you'll get an error. This is not the only method to get the bridge's ip address, but is probably the easiest. This function has no parameters/inputs.

Usage

```
get_hue_ip()
```

Bugs

Code repo: https://github.com/epijim/notifyme

Examples

```
## Not run: get_hue_ip()
```

get_light_info *Get info on lights connected to your Hue hub*

Description

This will return a dataframe containing information about the lights connected to your hue hub. It includes info on current, colour, is it on, and is it powered/reeachable.

Usage

```
get_light_info(bridge_ip = NULL, username = NULL,
  file = "~/r_keychain.rds")
```

Arguments

bridge_ip	Internal IP address of your hue bridge
username	Username for connecting to hue bridge
file	If the file listed here exists, it will try and use the private keys created by the function save_private_keys(). Just ignore if you want to manually give your bridge ip and username.

Bugs

Code repo: https://github.com/epijim/notifyme

Examples

```
## Not run: get_light_info(bridge_ip,username)
```

get_private_keys *Load a key from the keychain made by save_private_keys()*

Description

This function will look for a 'keychain' file, and if found load the key you asked for. Designed to be used with save_private_keys()

Usage

```
get_private_keys(api_var = "pushover_userkey", file = "~/r_keychain.rds")
```

Arguments

api_var	The name of the api key, this is user defined.
file	The name and location of the file where you want to store it. Default is same as the save function.

Bugs

Code repo: https://github.com/epijim/notifyme

Examples

```
## Not run: get_private_keys("keyImInterestedIn")
```

`hue_flashlights` *Flash all the lights connected to your hub*

Description

This function will flash the lights off and on a specified number of times. Currently, it will effect all lights connected to the hub.

Usage

```
hue_flashlights(bridge_ip = NULL, username = NULL, flashes = 10,
  flash_red = TRUE, light_name = NULL, file = "~/r_keychain.rds")
```

Arguments

`bridge_ip`	Internal IP address of your hue bridge
`username`	Username for connecting to hue bridge
`flashes`	Number of times to flash the lights on and off
`flash_red`	Do you want the lights to turn red before flashing?
`light_name`	If you want to flash ONE light, give it's name here as a character vector. Name is the actual name 'e.g. hallway', not the id number.
`file`	optional location of the keychain, if using

Bugs

Code repo: https://github.com/epijim/notifyme

Examples

```
## Not run: hue_flashlights(bridge_ip,username)
```

save_private_keys *Create a place to store API keys*

Description

This function will look for a 'keychain' file with your keys at the place you tell it to look with the file parameter. If it doesn't find it, it will make one.

Usage

```
save_private_keys(api_var = NULL, key = NULL,
  name_of_outputted_object = "api_keys", file = "~/r_keychain.rds")
```

Arguments

api_var	The name of the api key, this is user defined.
key	The actual key.
name_of_outputted_object	This is a convenience option, put in the name of the object you are assigning the output of the function to.
file	The name and location of the file where you want to store it. Default is unix home.

Details

This function WILL NOT actually save the file. Instead, it will return a dataframe with the keys, and give you the code to save the file to your system.

The intended use is store API keys in the home space.

Intended use

Save variables in a way that can automatically get read in by my other functions in this package. For the hue lights the variables I expect to see in api_var are "hue_ip" and "hue_username", while for pushover, the variables I expect to see are "pushover_userkey" and "pushover_apitoken".

See the example for a use example.

Bugs

Code repo: https://github.com/epijim/notifyme

Examples

```
## Not run: api_keys <- save_private_keys("new_key","THE KEY")
## Not run: # message returned is Run this code: saveRDS(api_keys, '~/r_keychain.rds')
## Not run: saveRDS(api_keys, '~/r_keychain.rds')
```

send_push_notification *Send push notification to devices*

Description

This function will send a push notification to your device via the push over API. You must make an account with that service (pushover.net) and get an API key and userkey.

Usage

```
send_push_notification(title = "Your R session says:",
  message = paste0("Message sent: ", Sys.time()), api_token = NULL,
  user_key = NULL, priority = "medium", file = "~/r_keychain.rds")
```

Arguments

title	Title of the push notification. Defaults to message from r.
message	Message body. Default just tells time message sent.
api_token	API token - create your own in a few minutes from pushover.net dashboard.
user_key	This is the key that identifies you. It's on the pushover.net dashboard.
priority	'low' means no beep/vibrate, 'medium' means beep/vibrate, 'high' means require response on device.
file	Optional - location of keychain if using.

Bugs

Code repo: https://github.com/epijim/notifyme

Examples

```
## Not run: send_push_notification(user_key = "xxxxxx", api_token = "xxxxx")
```

Index